Defense Technical Information Center
Compilation Part Notice

ADP014187

TITLE: Challenges of Aircraft Design Integration

DISTRIBUTION: Approved for public release, distribution unlimited

Availability: Hard copy only.

This paper is part of the following report:

TITLE: Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modelling and Virtual Simulation [La reduction des couts et des delais d'acquisition des vehicules militaires par la modelisation avancee et la simulation de produit virtuel]

To order the complete compilation report, use: ADA415759

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:
ADP014142 thru ADP014198

Challenges of Aircraft Design Integration

F. Kafyeke, M. Abdo, F. Pépin, P. Piperni, E. Laurendeau
Advanced Aerodynamics
Bombardier Aerospace
400 Côte Vertu Road
Dorval, Quebec, Canada, H4S 1Y9

Fassi.Kafyeke@notes.canadair.ca
Tel: (514) 855-7186

Abstract

The design of a modern airplane brings together many disciplines: structures, aerodynamics, controls, systems, propulsion with complex interdependencies and many variables. Recent aircraft programs, such as Bombardier's Continental Jet program (Figure 1) use participants located around the world and selected for their cost, quality and delivery capability. These participants share the risk on the program and must therefore be fully implicated in the design. A big challenge is to provide information on current design configuration simultaneously to all disciplines and all participants in the appropriate format. Another challenge of multidisciplinary optimization is to bring together technologies and methodologies of various disciplines in a way that is both practical and inclusive of the expertise that must accompany these individual technologies. This paper discusses progress made to address these challenges, streamline the aircraft design process and implement multidisciplinary optimization in an effective manner [1]. Initiatives include: implementation of the Bombardier Engineering System (BES) and of an MDO software environment (VADOR), linking of aerodynamic and structural design and analysis codes, validation of advanced wing design methods and calibration of viscous flow analysis and drag prediction methods.
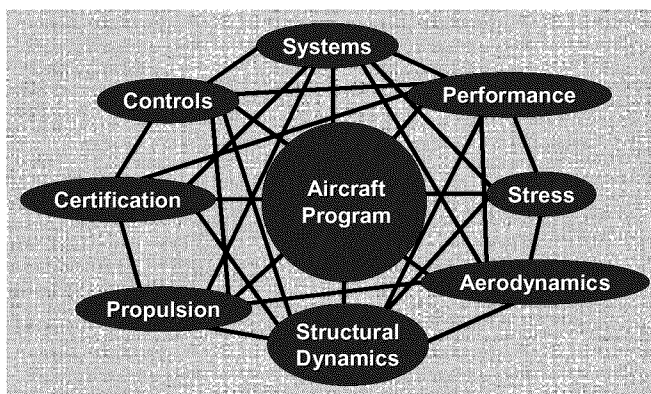
Figure 1:Bombardier Continental Business Jet

Figure 2: the complex interaction of various disciplines

Bombardier Engineering System (BES©)

The design of a modern airplane brings together many disciplines as illustrated in Figure 2: structures, aerodynamics, controls, systems, propulsion with complex interdependencies. The Bombardier Engineering System (BES©) was introduced to define clearly the phases, milestones and processes of an airplane design cycle and allow each process to be optimized. BES© describes how Bombardier defines, develops, certifies and validates commercial aerospace products. The system was implemented in response to various pressures on the aircraft design and development business. New and derivative aircraft have been launched almost every year in

the past dozen years. Many teams were involved in activities at multiple sites and for multiple programs and there was strong pressure from customers to reduce price, improve quality and deliver on time. On the basis of the company best practices, the roles and responsibilities of each engineering function were clearly defined. The first key elements of BES© are phases and milestones, illustrated in Figure 3. A phase is defined as a significant planned segment of a development project. Each project evolves through seven distinct phases (D1 to D7). A milestone is a planned event at a specific point in the project. It is not necessarily a single point in time. It is an opportunity to check progress and to evaluate plans. Management decision to proceed or not to proceed is determined at each milestone.
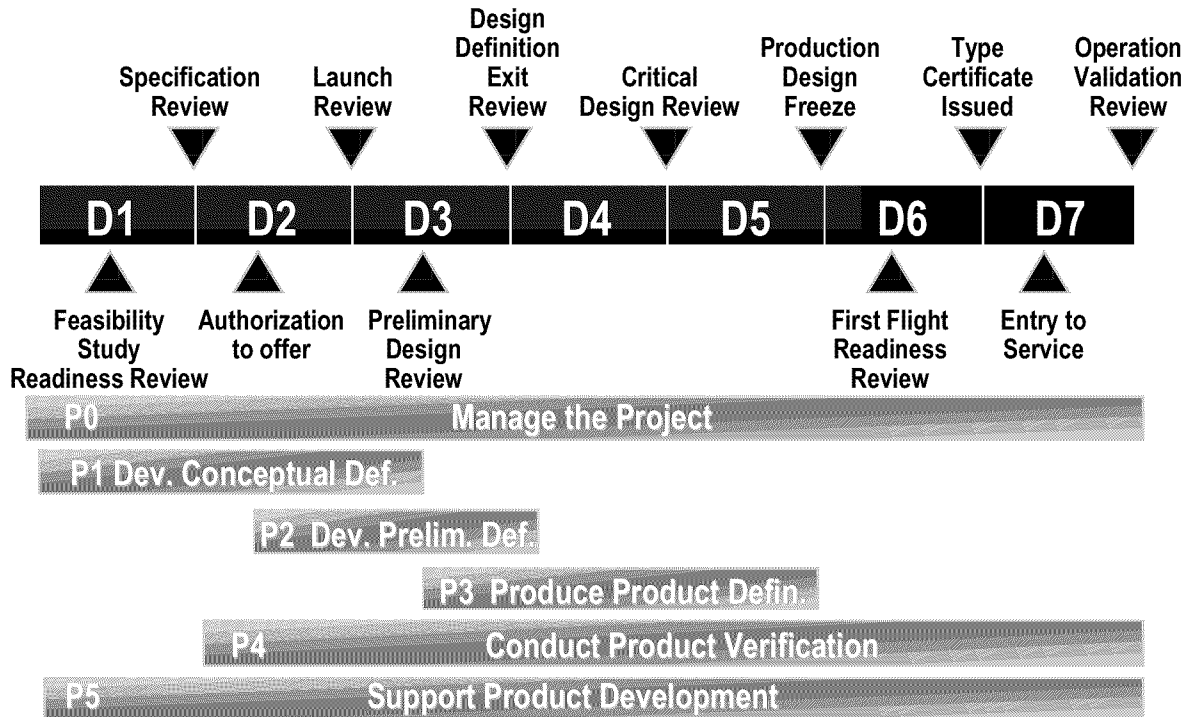

Figure 3: BES Phases, milestones and business processes

An important milestone is the decision to launch a new airplane, which occurs at the end of phase D2. At this point, the aerodynamic configuration must be frozen. This implies that most of the CFD and multidisciplinary design activity must take place in phases D1 and D2. The partners and suppliers of an aircraft program are brought together at two specific occasions. First, during the "Joint Conceptual Definition Phase" (JCDP, occurring in D1) where early configuration trade-offs take place. The second gathering occurs immediately after the aircraft formal launch, during the "Joint Definition Phase" (JDP, in D3).

The next important elements of BES are the product development processes. These are structured in a hierarchy, starting with five top business processes:

- Manage the Program and Project
- Develop the Conceptual Definition
- Develop the Preliminary Definition
- Produce the Product Definition
- Conduct Verification of the Product
- Support Product Development

Each business process is made of "Tier 2" Cross Functional Processes with deliverables associated to each process. A function's deliverable is a tangible data or a document required by an internal or external customer and used to monitor a project's progress. The next level of processes is the "Tier 3" level which describes

functional activities. A Tier 3 process map shows what activities are required to produce BES deliverables, what inputs are required to complete an activity, who supplies these inputs, who the customers for the deliverables are and the interactions with external entities such as vendors, partners or certification authorities. "Tier 4 Functional Tasks" are tasks required for preparing a specific deliverable. They are linked to a function's role and responsibility. Mapping Tier 4 functional tasks is a prerequisite to the wrapping of these tasks in a design automation software environment.

The BD-100 Continental Jet aircraft program is the first one to be developed entirely following BES procedures. The aircraft, shown on Figure 1, is designed to have a true north-American coast-to-coast capability with eight passengers. The overall design is geared towards shared ownership operations, requiring low cost, high utilisation, high dispatch reliability and good maintainability.

The aircraft technical specifications include:

- A maximum take-off weight of 37,500 lbs.;
- A range of 3,100 nautical miles NBAA/IFR with 8 passengers and baggage;
- A normal cruise speed of Mach 0.80 and a high cruise speed of Mach 0.82
- An initial cruise altitude of 41,000 ft and a maximum cruise altitude of 45,000 ft;
- A balanced field length below 5,000 ft;

The BD-100 program completed the Joint Conceptual Definition Phase before the formal program launch. This provided better definition of the aircraft early on. Partners and suppliers joined early. All functional groups participated in the design from the outset and all partners and suppliers used common design technology. Key milestone dates for the program so far are as follows:

- Market Debut   October 18 1998
- Joint Conceptual Definition   Aug 98 - May 99
- Full Program Launch   June 1999
- Joint Definition   June 99- Dec 99
- First Flight   August 14, 2001

Extensive cost trade-off parameters were prepared in all disciplines and all design decisions were subjected to an economic trade-off analysis. Weighing design proposals from vastly different disciplines is a challenging process. Only a fully integrated or collaborative multi-disciplinary optimization approach can guarantee the achievement of a true minimum of an overall aircraft level objective function. In reality, the merit of a design is heavily dependent on the experience and skill of the senior designers called to make the required decisions and on the experience of the design organisation as a whole.

On the multi-site and multi-partner BD-100 project, BES provided an effective method, to monitor and execute projects effectively, determine deliverables of the engineering process by function and harmonize product development processes and practices. BES offers two advantages. First, it helps clarify the company processes and deliverables. An objective analysis of these processes can then be made, possible deficiencies identified and corrected. Wherever possible, improved, more robust processes can be substituted. Secondly, because it requires process flowcharts down to the "Tier 4" functional activities, BES is a natural platform from which to establish integrated design procedures and automate them.

VADOR: an MDO Infrastructure Program

The design of a unique software framework tailored to Bombardier's Engineering System and capable of supporting Multi-Disciplinary Optimization (MDO) was entrusted to CERCA, a research center in numerical methods, located in Montreal. The software CERCA is developing is known as VADOR, for **V**irtual **A**irplane **D**esign and **O**ptimization **F**ramework. [2]. The central contribution of VADOR is the development of a software infrastructure permitting a seamless integration of technical applications and providing a global

perspective of a given design project. The software is capable of representing information, including data and methods, from the BES work flow charts to the detailed engineering tasks. It provides critical information on:

- The location of the data;
- The methods used to produce the data;
- The status of data and tasks;
- The validity of the data;
- The owners of the data and methods

The two main components of VADOR are "Data Components" and "Strategy Components". Data components are objects that encapsulate design and analysis data, usually contained in data files. A data component can encapsulate one or several data files. Strategy components are objects that encapsulate programs or analysis methodologies or processes. A program can be any piece of software that requires an input data and produces output data files. A process is a set of programs that must be executed in a sequence corresponding to a given algorithm. This sequence may include conditional loops. Both components have a set of attributes such as owner information, access permission, history, comments and present status. The framework allows collaboration and sharing of data and enforces proper documentation and promotes standardization of engineering methods. New processes and data are defined in the system during an integration phase. Subsequently, in a typical every day usage, the users simply execute these existing processes to create the required data and inspect the data using appropriate visualization tools provided through the framework. The clear separation between the data and the programs is intended to allow many different programs and processes to produce functionally equivalent data, reinforcing the standardization based on the data. This is crucial because the development and certification of a modern airplane requires extensive and rigorous documentation of all design characteristics.

Figure 4 illustrates the basic architecture of VADOR, composed of five elements, as described in [2]:

The **Graphical User Interface** (GUI) is a Java program running on the user's machine providing an interface between the engineer and the VADOR services.

The **Librarian Server** or Data server is a Java server program responsible for the handling and archiving of components.
The **Executive Server** is a Java server program that manages the execution of "Strategy Components" to create "Data Components". After receiving requests from the GUI to create data, it communicates with the Librarian server to retrieve data components and generates the data creation sequence according to the strategy invoked. The Executive server communicates with the CPU servers to run the required analysis programs and, after execution, notifies the Librarian Server to update the status of the components in the database.

The **CPU servers** are Java programs that wrap programs. They can be installed on any suitable machine. On request from the Executive server, they get the input files required for the execution, run the analysis programs and transfer the output files to the required locations.

The **Database Management System** (DBMS) is used to store, maintain and provide access information for components. The information is stored using a relational model. It should be noted that the framework does not manage detailed engineering data but rather references and information about the detailed data, stored in data files on the network.
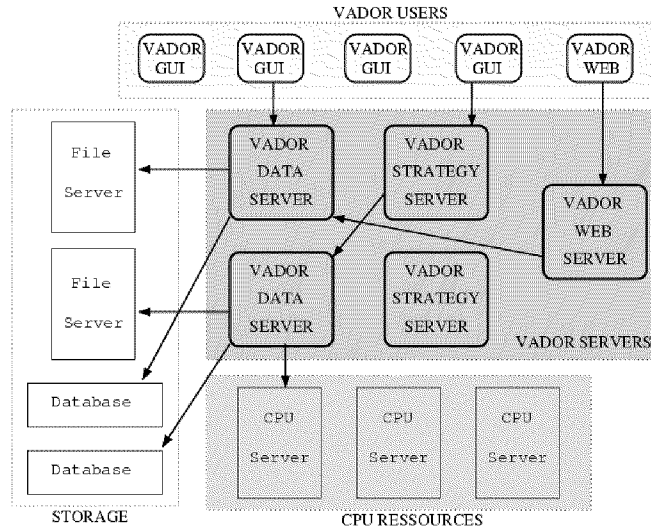
Figure 4: Typical VADOR Architecture

Aerodynamic analysis of transonic flexible wings

One important multi-disciplinary study is fluid-structure interaction. An initial objective was the prediction of wing weight and wing structural deformation (Figure 5) and the influence of this deformation on the aerodynamic load distribution. The prediction of the bending and twisting of wings was achieved by coupling the transonic CFD code KTRAN [3] with a thin-walled structural analysis program (TWSAP). The linear structural capabilities of the NASTRAN structural analysis software are utilized to predict the bending and twisting of a simplified finite element model (stick model) of the actual wing. Deformations predicted using stick models of transonic supercritical wings are in very good agreement with the results of full Finite Element Models (FEM) [4]. Results obtained for the static equilibrium (convergence) state of the Challenger and Global Express wings in 1g flight were found to be in very good agreement with experimental data. This suite of multi-disciplinary programs is wrapped in the VADOR framework.
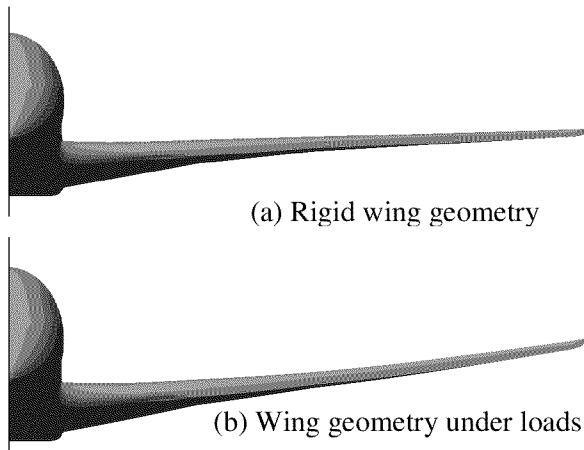

(a) Rigid wing geometry

(b) Wing geometry under loads

Figure 5: Static aeroelastic deformation of a transonic wing computed by the KTRAN /TWSAP /NASTRAN package at Mach 0.80 and CL=0.5

Wing Structure Conceptual Models
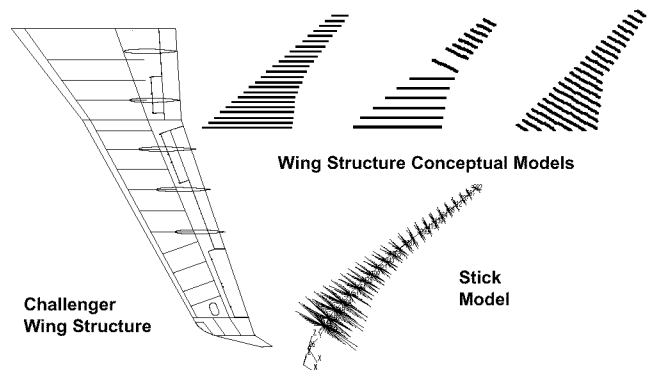
Challenger Wing Structure

Stick Model

Figure 6: Canadair Challenger wing structure and examples of conceptual structures generated by the TWSAP program

In a following step, the methodology was extended to predict the aeroelastic deformation of wings at the conceptual design stage. The program generates conceptual layouts of wing structural components and creates a beam finite element model of the wing structure (Figure 6). To establish the accuracy of the stick model designed by TWSAP, its prediction of the wing bending and twisting were compared with results obtained with the full finite element model of the real Challenger wing structure, without winglets. Figure 7 shows that the comparison obtained is sufficient for preliminary design purposes
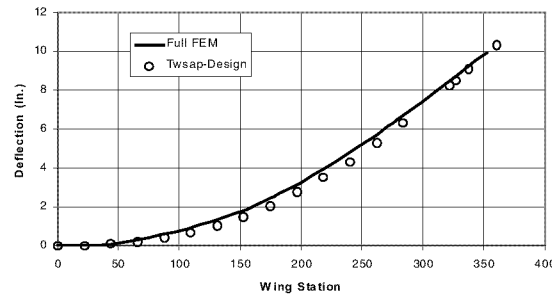

Figure 7: Comparison of wing bending predicted by the conceptual stick model and the full FEM of the Challenger wing without winglets.

Advanced aerodynamic wing design methods

To design wings, several methods are used. The most commonly used at Bombardier is the wing shape optimization program **ALLOP** developed in-house. Using a gradient-based optimizer, it is used to match a user-supplied target pressure distribution. The control points of a NURBS representation of the geometry are used as design variables in order to optimize the pressure distribution locally or globally. The ALLOP optimizer can call a variety of 2D and 3D high speed or low speed aerodynamic analysis codes. These codes return results that are used to calculate the current value of an objective function to minimize. Geometric constraints are imposed using penalty functions. Several developments were made to the representation of wing shape in order to include typical manufacturing constraints on the aerodynamic lines.

A second method, **INDES** [5, 6], an inverse design code originally developed by Tohoku University, in Japan, was linked to two transonic analysis codes. First, INDES was linked to the MGAERO 3D Euler code for complete aircraft configurations of Analytical Methods Inc. The MGAERO version used included a boundary-layer coupling introduced at Bombardier. INDES was also linked to the KTRAN transonic small disturbance code for complete aircraft configurations. This method can also be used to match a given target pressure distribution.

A third method, **AeroPointer** [7], recently licensed from Synaps Inc. of Atlanta, is an optimization environment capable of performing multi-disciplinary optimizations using a global parameter as an objective function, such as the total aircraft drag or weight. AeroPointer was linked to Bombardier's KTRAN transonic analysis code. The different capabilities of these methods are complementary, and each can be used effectively in the overall wing design process.

The main differences between ALLOP and INDES, the two methods for optimizing pressure distributions, is their relative speed of execution and flexibility. Since it is an inverse method, INDES is significantly faster. INDES will converge or achieve its best result in some 20 calls to the analysis code. In comparison, ALLOP requires hundreds of function calls, with the length of the optimization depending on the number of design variables. ALLOP optimizes the location of the control points of a NURBS representation of the geometry, so the more points are used or the greater the number of airfoil sections, the longer the optimization will last. INDES may or may not achieve a given target pressure distribution. If INDES does not converge on the specified target, the best that can be done is to modify the target pressure distribution itself. ALLOP is more

flexible because it can be restarted with a different set of design variables, and will usually continue to converge towards the target. Typically, for a complex design, ALLOP must be restarted a number of times, and the complete process may last in the order of two or three days. Unlike INDES, ALLOP will always produce smooth airfoil sections since it uses NURBS to describe the geometry. Another advantage of ALLOP is that it allows the user to work on a portion of a wing, or on a part of an airfoil section. For instance, the user may optimize only the upper surface of the wing, or only the leading edge, etc. For these reasons, INDES will typically be used to initiate an optimization process, because it does a good part of the work in a short period of time. ALLOP is then used to refine the design.

The major disadvantage of using either ALLOP or INDES is the requirement to define a target pressure distribution. This is not only a time consuming process but it also assumes that the designer has enough experience to "know" what an optimal pressure distribution is for a specific wing. In contrast, no target pressure distribution is required when using the AeroPointer software. The latter is capable of performing a multi-disciplinary optimization by minimizing a global parameter, such as a combination of the total drag and the weight of an aircraft. This capability is very useful since it makes no assumptions about the pressure distribution, and it effectively automates the design process.

Automation in the design process is important not only from the point of view of efficiency, but also because it makes multi-disciplinary optimization possible, since the latter can only be done through the minimization of global parameters. AeroPointer achieves this capability through the use of a hybrid optimizer that combines the capabilities of genetic, gradient, and simplex methods. AeroPointer also allows the user to define any geometric parameter as a design variable or a constraint, and can perform weighted multi-point optimizations. Naturally, the quality of the final design will depend on the fidelity of the analysis code and on the topology of the design space. Typically, a careful selection of design variables and constraints is required to ensure a successful optimization, and the methodology developed for one application may not necessarily be optimal for another. In some cases, AeroPointer will produce a good design but one that can clearly be improved in some areas. In such a case, an AeroPointer optimization would be followed by the further optimization of the pressure distributions using either INDES or ALLOP.

The current best approach therefore is to initiate the design process with AeroPointer to define the general characteristics of the optimal design in an MDO sense (Figure 8). This process would typically be initiated with a low fidelity analysis code and completed with a higher fidelity code whenever practical. Once the general characteristics of the configuration have been defined, any further improvements required in the pressure distributions could then be achieved using INDES if possible or ALLOP when detailed refinements are required (Figure 9).
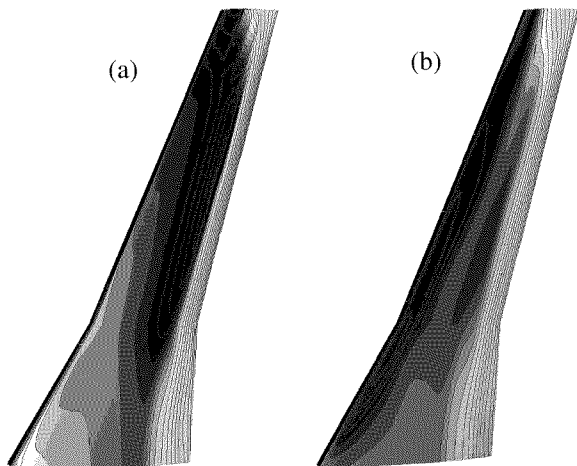

Figure 8: (a) Initial business jet configuration; KTRAN solution; M=0.8 CL=0.5. (b) Configuration optimized with AeroPointer/KTRAN; M=0.8 CL=0.5
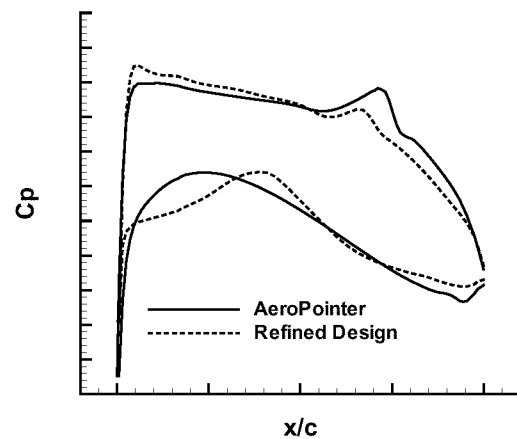
Figure 9: Pressure distribution achieved with an MDO sense optimization compared to a final design including a locally refined pressure distribution.

CFD flow analysis and drag prediction

Bombardier recent CFD development efforts have concentrated on the Full Aircraft Navier-Stokes Code FANSC [8]. The program uses multi-block structured grids, with an unstructured block topology, i.e. it allows any number of blocks to merge at the same location. It uses a cell-centered finite volume approach with a choice of space-discretization schemes and an explicit Runge-Kutta time-marching method. The code can be run in Euler mode, in Euler mode with boundary-layer coupling and in Navier-Stokes mode. The Navier-Stokes code uses the Spalart-Allmaras turbulence model [9].

To be useful in a realistic design environment, the FANSC code was made robust for solutions on complex aircraft geometry. Boundary conditions include no-slip and slip walls, transpiration wall for boundary-layer coupling, symmetry and degenerate lines and points, Riemann and engine inlet/outlet boundary. The code allows also the specification of multiple boundary conditions on each block face. Its run time efficiency was considerably improved by adding coarse grain parallelization on blocks (3.6 out of 4 CPUs) and vectorization (94% efficient).

The large CPU time of Navier-Stokes computations still precludes their inclusion in routine design and optimization loops. Euler/boundary-layer computations are used instead. A boundary-layer code was developed and coupled with FANSC first through the use of a direct Viscous/Inviscid Interaction (VII) scheme [10]. The coupling uses a transpiration velocity approach, with no need to regenerate a new mesh at every VII cycle. Since a direct VII procedure fails when separated flow is encountered, as often found during design iterations, an inverse boundary-layer code was also coupled with FANSC using a quasi-simultaneous VII scheme. The viscous flow is solved with the CIBL3D inverse code, developed by Cebeci et al. at California State University, Long Beach [11]. With this code, separated boundary layers can be computed with accuracy comparing favorably with more time-consuming Navier-Stokes computations for many cases of interest. This is illustrated in Figure 10, showing a pressure distribution computed at mid-span of a Challenger wing-body configuration.
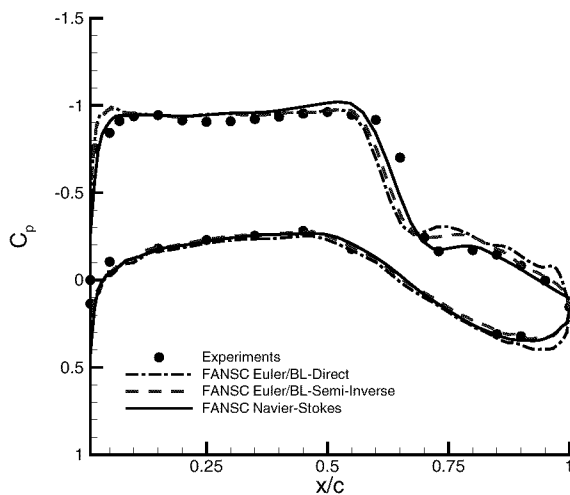

Figure 10: Euler/Boundary layer and Navier-Stokes computations of flow on the Challenger wing/body configuration Mach 0.82, Alpha = 1.5 degrees, Rec = 6 Million, station at 40.5% of the wing semi-span
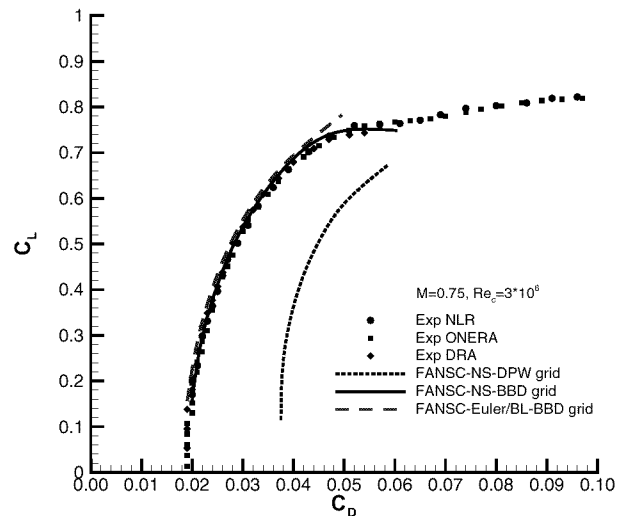
Figure 11: FANSC prediction of drag polar for the DLR-F4 configuration, Mach 0.75, Reynolds number 3 Million.

To be used effectively in aerodynamic design loops, CFD codes must produce accurate, reliable and repeatable drag estimates. Drag modules were constructed as a post-processing step to the Euler/Boundary-layer solutions. They include a semi-empirical module for fuselage and nacelle drag, a Multhopp algorithm for induced drag, Lock's method for computation of wave drag based on shock strength and a Squire-Young

module for the computation of wing and tailplane viscous drag. Far-field methods for the induced drag and the computation of wave drag from the integration of entropy variation across shock waves were also investigated. More recently, investigations were made in the prediction of drag from direct integration of pressures and skin friction obtained with a high-accuracy Navier-Stokes solution.

The difficulties of drag prediction with Navier-Stokes computations were illustrated at an AIAA Drag Prediction Workshop held in June 2001 in Anaheim, California [12]. FANSC was used to predict the drag polar of a DLR-F4 wing-body for which experimental results had been collected in NLR, ONERA and DRA wind tunnels. Drag was obtained from integration of pressure and skin friction coefficients, as specified for the workshop. Initial predictions made by FANSC with the grid supplied by the workshop organizers showed drag levels much higher than experimental values (DPW grid results, in Figure 11). A new mesh of the DLR-F4 configuration generated using Bombardier's MBGRID program was prepared. The mesh had good orthogonality on the solid surfaces, 10^{-6} chord wall spacing, 3.8 Million mesh points, an open wing tip and a blunt trailing edge. Calculations with the same program on this mesh showed excellent correlation with the experimental values (BBD grid results, in Figure 11). The main difference between the two grids was in the orthogonality near solid surfaces (Figure 12). FANSC showed excellent convergence characteristics (density and turbulent viscosity) on the Bombardier generated grid and on the workshop supplied grid, despite its excessive skewness. Integrated lift and pitching moment predictions on the workshop grid were accurate. Mesh skewness introduced discretization errors on the skin-friction evaluations whereas pressure drag was correctly predicted on both grids. This illustrates the great care that must be exercised if drag from Navier-Stokes computations is used as the function to be minimized in a wing design process. One must ensure that mesh modifications do not introduce variations not due to the wing geometry changes. Figure 11 shows that an excellent drag polar was also obtained on the NLR-F4 configuration with FANSC running in Euler/boundary-layer mode with the post-processing drag formulas. The grid required for this calculation was much simpler, with 1.3 million grid points. A solution for one angle of incidence is obtained in 0.45 hours on 8 CPUs of a Cray SV1 computer instead of the 6 hours required by the Navier-Stokes calculation. The Euler solution required 600 Mbytes memory instead of the 2.2 Gbytes required by the Navier-Stokes analysis. There are therefore advantages in using Euler/boundary-layer methods in large parts of the wing design process.
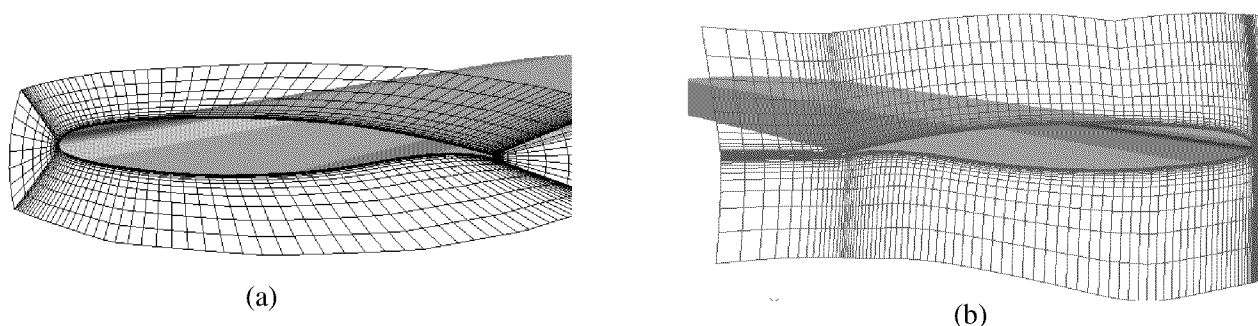

(a)

(b)

Figure 12: Close-up views of DLR-F4 grids for the AIAA 2001 Drag Prediction Workshop. (a) Initial supplied grid; (b) Bombardier-generated grid.

Despite considerable progress made to date, the use of Navier-Stokes methods in aircraft design integration is still a challenge. The best approach seems to be the use of a full suite of low and high fidelity codes, starting for instance with low fidelity codes and finishing with the more sophisticated methods. The goal is to implement Navier-Stokes codes in design loops in the coming year, following a significant upgrade of the available computing hardware.

Conclusions

The toughest challenge of MDO in the aerospace industry is to bring together technologies and methodologies of various disciplines in a way that is both practical and inclusive of the expertise that must accompany the individual technologies. MDO procedures asking experienced engineers to change completely their methods

have often met with mixed success. Our approach, described here, is to develop this capability in a step-by-step approach that includes either the actual methods in use by the various disciplines, or simplified versions of these methods. Bombardier's approach to automated design integration includes the following steps:

- Understanding, documention and optimization of all engineering procedures required to develop an aircraft (BES)
- Implemention of a software design environment which erases geographical distance and differences in computing platforms and automates BES tier 4 processes (VADOR)
- Implemention and validation of multi-disciplinary analysis and optimization procedures (fluid-structure, aerodynamics and systems simulation, etc.)

Acknowledgments

References

1. W. Elliot and B. Leigh, "Aircraft Loads Methodology For MDO", AIAA-2001-1432.
2. Alzubbi, A. Ndiaye, B. Mahdavi, F. Guibault, B. Ozell and J.-Y. Trépanier, "On the use of JAVA and RMI in the development of a computer framework for MDO, AIAA Paper 2000-4903
3. F. Kafyeke, P. Piperni and S. Robin, "Applications of KTRAN Transonic Small Disturbance code to the Challenger Business Jet Configuration with Winglets", SAE Paper 881483, October 1988.
4. M. Abdo, F. Kafyeke, F. Pépin, Z. Borowiec and A. Marleau, "Transonic Aerodynamics of Flexible Wings", CASI 48th Annual Conference Proceedings, April 2001.
5. Obayashi, S., Takahashi, S. and Fejtek, I., "Transonic Wing Design by Inverse Optimization Using MOGA", CFD98, 6th Annual Conference of the CFD Society of Canada, Quebec City, June 1998.
6. D. Jones, I. Fejtek, and D. Leblond, "Coupling of a Wing Inverse Design Code to a Transonic Small Disturbance Flow Solver", CASI 48th Annual Conference Proceedings, April 2001.
7. Van der Velden, "Aeropointer, a commercial CFD shape optimization tool", CASI 48th Annual Conference Proceedings, April 2001.
8. Zhu, Z., Laurendeau, E. and Mokhtarian, F., "Cell-Centered and Cell-Vertex Algorithm for Complex Flow Configurations", 8th Annual Conference of the CFD Society of Canada, Montreal, June 2000.
9. Spalart, P.R. and Allmaras, S.R., "A One-Equation Turbulence Model for Aerodynamic Flows", AIAA Paper 92-0439, Jan. 1992.
10. E. Laurendeau and F. Mokhtarian, "Coupled Euler/Boundary-Layer Methods with the FANSC Code", CASI 48th Annual Conference Proceedings, April 2001.
11. Cebeci, T., Jau, J. and Vitiello, D. "An Interactive Boundary-Layer Approach to Multi-Element Airfoils at High Lift", AIAA Paper 92-0404, January 1992.
12. AIAA CFD Drag Prediction Workshop Website. http://www.aiaa.org.tc/apa/dragpredworkshop/dpw.html, June 2001

Paper #48
Discussor's Name: D. Lovell
Author's Name: Dr. F. Kafyeke

Q: You have stated that external shape fixed in stages 1 & 2. What do you do about ensuring internal items can be packaged within the external shape?

A: We try and examine structural and systems implications as early as possible in the design stages (D1, D2) hence the value of MDO at conceptual design stage. If this is well done, only <u>minor</u> modifications to the external shapes are needed at D2.